

DL4G - Questionnaire

Deep Learning for Games

Maurin D. Thalmann

19. Januar 2020

Dieser Questionnaire wurde basierend auf einer Card2Brain Sammlung erstellt:
Card2Brain - DL4G (Credits: Cyrille Ulmi)

Inhaltsverzeichnis

1	Sequenzielle Spiele	2
1.1	Was sind die Eigenschaften von endlichen-sequenziellen Spielen?	2
1.2	War wird unter Perfect Recall verstanden?	2
1.3	Was ist eine Strategie?	2
1.4	Was ist ein Strategie-Profil?	2
1.5	Was ist eine Utility- oder Payoff-Function?	2
1.6	Was sind die Komplexitätsfaktoren bei einer Spielanalyse?	2
1.7	Was ist imperfekte Information?	2
1.8	Beispiele von Spielen mit perfekten / imperfekten Informationen?	2
1.9	Was ist der Suchraum?	2
1.10	Was ist ein Suchbaum?	2
1.11	Wie funktioniert Backward Induction?	3
1.12	Was bedeutet Rationalität?	3
1.13	Welche Arten von Lösungen werden bei endlich-sequenziellen Spielen unterschieden?	3
1.14	Was versteht man unter einem Zero-Sum Game (Nullsummenspiel)?	3
1.15	Was sind Charakteristiken des Minimax-Algorithmus?	3
1.16	Wie funktioniert der Minimax-Algorithmus?	3
1.17	Was versteht man unter Search Tree Pruning?	3
1.18	Was sind die Regeln von Alpha-Beta Pruning?	3
1.19	Was ist der Vorteil von Alpha-Beta Pruning?	4
2	Monte Carlo Tree Search	4
2.1	Wieso werden Random Walks eingesetzt? (Tree Search)	4

1 Sequenzielle Spiele

1.1 Was sind die Eigenschaften von endlichen-sequenziellen Spielen?

- Eine endliche Anzahl Spieler mit einer endlichen Anzahl Aktionen
- Die Aktionen werden sequenziell ausgewählt
- Es wird eine endliche Anzahl Runden gespielt
- Spätere Spieler sehen die Aktionen vorheriger Spieler

1.2 Was wird unter Perfect Recall verstanden?

Perfekte Erinnerung an alle vorherigen Züge

1.3 Was ist eine Strategie?

Sagt einem Spieler, welche Aktion im aktuellen Zug auszuführen ist

1.4 Was ist ein Strategie-Profil?

Die ausgewählte Strategie eines Spielers

1.5 Was ist eine Utility- oder Payoff-Funktion?

Sie berechnet das Resultat für jede Aktion

1.6 Was sind die Komplexitätsfaktoren bei einer Spielanalyse?

- Anzahl Spieler
- Grösse des Suchraums (Anzahl gespielte Züge & Anzahl mögliche Aktionen)
- Kompetitiv vs. Kooperativ
- Stochastische Spiele (mit Zufall) vs. Deterministisch
- Perfekte vs. imperfekte Information

1.7 Was ist imperfekte Information?

- Das Spiel konnte nur teilweise beobachtet werden
- Man kennt bspw. nicht die Karten der anderen Spieler

1.8 Beispiele von Spielen mit perfekten / imperfekten Informationen?

Perfekt (Schach) und imperfekt (Jass, Poker)

1.9 Was ist der Suchraum?

Anzahl gültige Brettpositionen und die untere Grenze des Suchbaums

1.10 Was ist ein Suchbaum?

- Knoten sind Spielpositionen / Spielzustände
- Kanten sind Aktionen / Spielzüge
- Blätter werden durch Payoff-Funktionen definiert

1.11 Wie funktioniert Backward Induction?

- Den Baum von unten nach oben durcharbeiten (bzw. von rechts nach links)
- Immer den besten Weg für den aktuellen Spieler markieren
- Geeignet für sequenzielle endliche Spiele mit perfekter Information

1.12 Was bedeutet Rationalität?

Dass der Spieler nicht die schlechtere Alternative wählt

1.13 Welche Arten von Lösungen werden bei endlich-sequenziellen Spielen unterschieden?

- Ultra-schwache Lösung
 - Bestimmt, ob der erste Spieler einen Vorteil aus der Initialposition hat, ohne die genaue Strategie zu kennen
 - Setzt perfektes Spielen des Gegners voraus
 - Beispielsweise durch Existenzbeweise in der Mathematik
- Schwache Lösung
 - Kann ein komplettes Spiel mit perfekten Zügen aus der Initialposition durchspielen
 - Geht von einem perfekten Spiel des Gegners aus
- Starke Lösung
 - Kann aus jeder Position heraus perfekte Züge spielen
 - Kann auch gewinnen, wenn vorherige Spieler einen Fehler gemacht haben

1.14 Was versteht man unter einem Zero-Sum Game (Nullsummenspiel)?

- Der Vorteil für einen Spieler ist zum Nachteil des anderen Spielers
- Die Punktesumme für zwei Strategien ist immer gleich Null

1.15 Was sind Charakteristiken des Minimax-Algorithmus?

- Gilt nur für ein Nullsummenspiel
- Zwei Möglichkeiten / Ziele
 - den eigenen Gewinn maximieren
 - den Gewinn des Gegners minimieren

1.16 Wie funktioniert der Minimax-Algorithmus?

- Wenn der Knoten mir gehört: Aktion wählen, die den Payoff maximiert
- Wenn der Knoten dem Gegner gehört: Aktion wählen, die den Payoff minimiert
- Wenn es ein Endknoten ist: den Payoff berechnen

1.17 Was versteht man unter Search Tree Pruning?

Nicht relevante Teilbäume können weggelassen werden, reduziert den Rechenaufwand

1.18 Was sind die Regeln von Alpha-Beta Pruning?

- α ist der grösste Wert aller MAX Vorfahren eines MIN Knoten
- β ist der kleinste Wert aller MIN Vorfahren eines MAX Knoten
- Den Teilbaum abschneiden, falls er grösser als α oder kleiner als β ist

1.19 Was ist der Vorteil von Alpha-Beta Pruning?

- b = Anzahl Kanter der Knoten und m = Tiefe des Baums
- Ordnung verbessert sich von $O(b^m)$ nach $O(b^{m/2})$, halbiert also die Tiefe der Suchbäume

2 Monte Carlo Tree Search

2.1 Wieso werden Random Walks eingesetzt? (Tree Search)

-